COBB

INSIDE DOS

December 1993
Vol. 4 No. 12

Tips & techniques for MS-DOS & PC-DOS Versions 5 & 6

Creating a holiday screen saver

VERSIONS
5.0 & 6.0

Do you need something to help you get into the spirit of the season? How about some holiday music or a little snow? If a good dose of holiday cheer is in order, a QBasic holiday screen saver is sure to fill the bill. In this article, we'll show you how to create our LETITSNO screen saver. The musical introduction, set to the tune of "Let It Snow!," is sure to put a smile on your face, and the steady snowfall that follows provides temporary relief from the often hectic pace of the holiday season.

Creating the program

Since LETITSNO is a QBasic program, you create it by using the QBasic program editor. To start the editor, type

```
C:\>qbasic c:\dir\letitsno
```

where *dir* is the directory you want to store the program in. Then, press [Enter]. A blank document screen will appear, with the program filename LETITSNO.BAS in the window title bar.

Next, type the program shown in Figure A on pages 2 and 3, pressing [Enter] to end each line. As you enter the code, don't be concerned about matching the upper- and lowercase letters in Figure A. QBasic will take care of converting the necessary keywords into uppercase letters.

QBasic will also try to catch syntax errors by prompting you if you try to leave the current line without completing it in the proper syntax. For example, suppose you type the line

```
dim snowcolumn%(75
```

and press [Enter]. Immediately, QBasic displays the prompt shown in Figure B on page 3 and highlights the location of the error—the proper syntax for this line calls for a closing parenthesis. To dismiss a QBasic error prompt and return to the error location, click <OK> or press [Enter].

After entering the program and checking for typing errors, issue the Save command from the File menu ([Alt]F,S). You're now ready to run LETITSNO.

Running the LETITSNO screen saver

From the QBasic program editor, run LETITSNO by pressing [Shift][F5]. First, the screen clears, switches to graphics mode (320 x 200 resolution), and turns light blue. Then, as the melody of "Let It Snow!" plays, the lyrics of the first verse appear onscreen in sync with the appropriate tones. As the last line of the melody plays, the phrase *LET IT SNOW!* appears and falls like snow, as shown in Figure C on page 3. On the final note of the tune, snowflakes begin to fall, gathering at the bottom of the screen, as shown in Figure D, also on page 3.

As the program progresses, the snow falls shorter and shorter distances as it piles up. Once the snow reaches a point where it can't fall anymore, the screen clears and a new series of snowfalls begins.

To stop the program, simply press a key. The program will end and display the message *Press any key to continue* at the bottom of the screen as soon as the current snowfall reaches the ground. (If you press a key during the musical introduction, the program will continue until the first snowfall reaches the ground.) Pressing a key again returns you to the LETITSNO.BAS screen in the QBasic program editor. To return to the DOS prompt, issue the Exit command from the File menu ([Alt]F,X).

IN THIS ISSUE

Figure A

```
'LETITSNO.BAS--This program is a holiday screen saver.

DIM snowcolumn%(75)                                    'sets # of falling snowflakes
DIM snowrow%(75)                                       'sets vertical spread of falling snowflakes
snowlimit% = 124                                       'initializes snow level

CLS                                                    'clears screen
SCREEN 1                                               'switches video adapter to 320 x 200 graphics mode
COLOR 9                                                'sets screen background color to light blue
RANDOMIZE TIMER                                        'turns on random number generator

FOR currentrow% = 1 TO 3                               'sets current text line
    FOR dataset% = 1 TO 8                              'specifies current data set
    READ column%, phrase$, frequency%, duration%       'reads starting position, text, note, length of note
    SOUND frequency%, duration%                        'plays note
    FOR delay% = 1 TO 20000                            'creates delay between each note for clarity
    NEXT delay%
    LOCATE currentrow%, column%                        'moves text cursor to current line, starting position
    PRINT phrase$                                      'prints text
    NEXT dataset%                                      'moves data pointer to next data set
NEXT currentrow%                                       'moves to next text line

FOR delay1% = 1 TO 5                                   'creates delay before last line of song
    FOR delay2% = 1 TO 8000                            'extends delay
    NEXT delay2%
NEXT delay1%

FOR lastset% = 1 TO 9                                  'specifies current data set
    READ column%, phrase$, frequency%, duration%
    SOUND frequency%, duration%
    FOR wordfall% = 5 TO 9                             'sets range of rows for text to fall
    LOCATE wordfall%, column%                          'moves text cursor to current line, starting position
    PRINT phrase$                                      'prints text
    LOCATE wordfall% - 1, column%                      'moves text cursor to previous line, starting position
    PRINT SPACE$(LEN(phrase$))                         'erases text on previous line to create illusion of falling
    FOR delay% = 1 TO 7000                             'creates delay to slow down fall
    NEXT delay%
    NEXT wordfall%
```

LETITSNO.BAS is a QBasic holiday screen saver program.

INSIDE DOS

Inside DOS (ISSN 1049-5320) is published monthly by The Cobb Group.

Prices

Domestic:	$49/yr ($6.00 each)
Outside US:	$69/yr ($8.50 each)

Phone

Toll free:	(800) 223-8720
Local:	(502) 491-1900
Customer Relations Fax:	(502) 491-8050
Editorial Department Fax:	(502) 491-4200

Address You may address tips, special requests, and other correspondence to

The Editor, *Inside DOS*
9420 Bunsen Parkway, Suite 300
Louisville, Kentucky 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, Kentucky 40220

Copyright Copyright © 1993, The Cobb Group. All rights reserved. *Inside DOS* is an independently produced publication of The Cobb Group. The Cobb Group reserves the right, with respect to submissions, to revise, re-publish, and authorize its readers to use the information submitted for both personal and commercial use.

The Cobb Group, its logo, and Satisfaction Guaranteed statement and seal are registered trademarks of The Cobb Group. *Inside DOS* is a trademark of The Cobb Group. Microsoft and MS-DOS are registered trademarks and Microsoft Windows is a trademark of Microsoft Corporation. PC-DOS is a trademark of IBM Corporation.

Postmaster Second class postage is pending in Louisville, KY. Send address changes to

Inside DOS
P.O. Box 35160
Louisville, KY 40232

Authorized Canada Post International Publications Mail (Canadian Distribution) Sales Agreement #XXXXXX CANADA GST #123669673. Send returns to Canadian Direct Mailing Sys. Ltd., 920 Mercer Street, Windsor, Ontario, N9A 7C2. Printed in the USA.

Staff

Editor-in-Chief:	Charity Edelen
Contributing Editor:	Van Wolverton
Editing:	Elizabeth Welch
	Timothy E. Hampton
Production Artist:	Julie Jefferson
Design:	Karl Feige
Publications Manager:	Tara Dickerson
Managing Editor:	Suzanne Thornberry
Circulation Manager:	Brent Shean
Publications Director:	Linda Baughman
Editorial Director:	Jeff Yocom
Publishers:	Mark Crane
	Jon Pyles

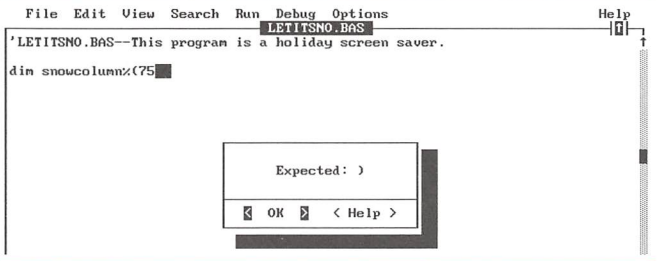
Advertising: For information about advertising in Cobb Group journals, contact Tracee Bell Troutt at (800) 223-8720, ext. 430.

Back Issues To order back issues, call Customer Relations at (800) 223-8720. Back issues cost $6.00 each, $8.50 outside the US. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all articles that have appeared in *Inside DOS*.

Advisory Board Earl Berry Jr.
Tina Covington
Marvin D. Livingood

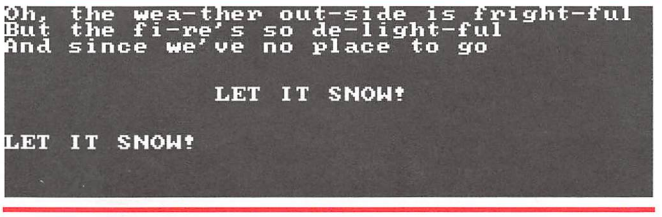
```
NEXT lastset%                                                    'moves data pointer to next data set

FOR delay% = 1 TO 20000                                          'creates delay before snowfall
NEXT delay%

DO
    FOR flake% = 1 TO 75                                         'sets counter for falling snowflakes
        snowcolumn%(flake%) = INT(RND(1) * 320)                  'sets random column for snowflake
        snowrow%(flake%) = INT(RND(1) * 75)                      'sets random location of snowflake within vertical spread
    NEXT flake%                                                  'prepares next falling snowflake
    FOR row% = 1 TO snowlimit%
        FOR flake% = 1 TO 75
            CIRCLE (snowcolumn%(flake%), snowrow%(flake%) + row%), 1     'draws current falling snowflake
        NEXT flake%
        FOR flake% = 1 TO 75
            CIRCLE (snowcolumn%(flake%), snowrow%(flake%) + row% - 1), 1, 0   'erases flake on previous line to create falling illusion
        NEXT flake%
        FOR randomflake% = 1 TO 30                               'sets counter for filler snowflakes at ground level
            nosnow% = INT(RND(1) * (snowlimit% - 75))            'sets upper level for filler snowflakes
            CIRCLE (INT(RND(1) * 320), 200 - nosnow% + row%), 1  'draws filler snowflake
        NEXT randomflake%                                        'prepares next filler snowflake
        CIRCLE (INT(RND(1) * 320), INT(RND(1) * 200)), 1         'creates random floating flakes
    NEXT row%                                                    'moves to next row
    snowlimit% = snowlimit% - 1                                  'raises snow level
    IF snowlimit% = 0 THEN                                       'checks if snow level is at top of screen
        snowlimit% = 124                                         'if so, resets snow level
        CLS                                                      '      and clears screen to start snow pileup over
    END IF
LOOP UNTIL INKEY$ <> ""                                          'repeats snowfall until user presses a key

DATA 1, "Oh, the", 147, 9, 9, "wea-ther", 294, 9, 18, "out-", 262, 9, 22, "side", 247, 9, 27, "is", 220, 9, 30, "fright-", 196, 9, 37, "ful", 147,
    17, 40, "", 32000, 1
DATA 1, "But the", 147, 9, 9, "fi-", 220, 14, 12, "re's", 196, 4, 17, "so", 220, 14, 20, "de-", 196, 4, 23, "light-", 185, 9, 29, "ful", 147, 17, 40,
    "", 32000, 1
DATA 1, "And", 165, 9, 5, "since we've", 330, 9, 17, "no", 294, 9, 20, "place", 262, 9, 26, "to", 247, 9, 29, "go", 220, 19, 32, " ", 220, 1, 40, "",
    32000, 1
DATA 1, "LET IT SNOW!", 370, 5, 13, " ", 330, 1, 14, "", 294, 8, 14, "LET IT SNOW!", 294, 5, 26, " ", 262, 1, 27, "", 247, 8, 27, "LET IT SNOW!!!",
    247, 5, 40, "", 220, 1, 40, "", 196, 21
```

Figure B



When you use the incorrect syntax for a QBasic statement, the program editor displays an error prompt.

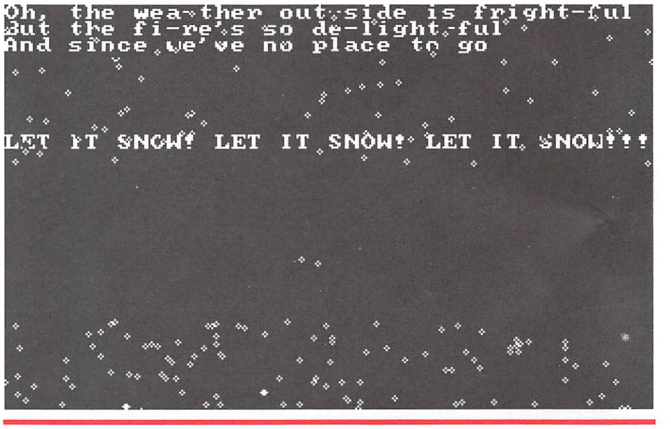
Figure C



The LETITSNO screen saver features falling lyrics in sync with the last line of the melody.

Note: The speed at which QBasic executes programs might vary from machine to machine. If the melody seems to play too slowly or out of sync with the lyrics, try decreasing the upper limits in the FOR delay*x* statements (those shown in red in Figure A).

Figure D



In addition to using text animation, LETITSNO uses graphics animation to create the illusion of falling snow.

Running LETITSNO from the DOS prompt

Once you've created the LETITSNO screen saver, you'll probably want to run it before you leave your computer for an extended period. You can run it directly from the

DOS command line without pressing [Shift][F5] from the QBasic program editor. At the prompt, you simply type

```
C:\>qbasic /run c:\dir\letitsno
```

and press [Enter]. If the directory in which you saved LETITSNO.BAS appears in the PATH statement of your AUTOEXEC.BAT file, you can run LETITSNO from any directory without specifying the path.

Again, you press a key to stop the program. When you press a key in response to the *Press any key to continue* prompt, the program returns you to the LETITSNO.BAS screen in the QBasic program editor.

Handling runtime errors

If an error occurs while you're running the LETITSNO.BAS program, QBasic will end execution of the program, return to the program editor, display a prompt describing the error, and highlight the line the error occurred in. To dismiss the error prompt, click <OK> or press [Enter]. Then, check the highlighted line for errors. If you don't find any, check the lines above the one that's highlighted. Occasionally, a typing error in a previous line might cause an error to occur later in the program.

LETITSNO on ZiffNet

As an added benefit to our readers, all the batch files and programs we've published in *Inside DOS*—including LETITSNO.BAS—are available through The Cobb Group Online on ZiffNet. See "Downloading Files from The Cobb Group Online" below for instructions on downloading these files. LETITSNO is also available on this month's DOS Software Connection.

THE COBB GROUP ONLINE

VERSIONS 5.0 & 6.0

Downloading files from The Cobb Group Online

As we mention in our lead article, "Creating a Holiday Screen Saver," all batch files and programs that appear in *Inside DOS* are available from The Cobb Group Online on ZiffNet. The Cobb Group Online is an easily accessible source of these programs as well as programs that appear in other Cobb Group journals. Created by Ziff Communications, ZiffNet is a comprehensive online information service that brings together the best in downloadable freeware and shareware, discussion forums, computing news, product and service reviews, and industry analysis.

ZiffNet services cost $2.50 per month plus connect-time charges of $12.80 per hour for 1200 or 2400 BPS, or $22.80 per hour for 9600 BPS. If you have a modem, the information in this article will help you log onto ZiffNet and download files from The Cobb Group Online.

Logging onto ZiffNet with a CompuServe ID

If you're already a CompuServe subscriber, you can access The Cobb Group Online on ZiffNet from any CompuServe! prompt. You simply type *go znt:cobbapp*.

Logging onto ZiffNet if you aren't a CompuServe subscriber

If you don't subscribe to CompuServe, you must get a ZiffNet ID to access The Cobb Group Online. First, find out your local access number by calling (800) 635-6225. By using this local access number, you won't have to pay any long-distance charges to log onto ZiffNet.

Once you have your local access number, you're ready to connect to ZiffNet. Set your communications software to 7 data bits, even parity, one stop bit. Then, select a transfer rate of 1200, 2400, or 9600. Next, use your modem to dial your local access number. Once you're connected, press [Ctrl]C and respond to the prompts by entering

```
User ID: 177000,5555
Password: ZIFF*NET
```

At the *Select a Language for Signup* prompt, enter an appropriate choice. Then, at the *Enter Agreement Number* prompt, type *cobbapp*.

A series of prompts will follow, asking you for such information as your name and your credit card number for billing purposes. Once you've entered the requested information, ZiffNet will assign you a user ID and a temporary password. Make sure you write these down—you'll need them to log on to Ziffnet during the next ten days. Within this time, you'll receive in the mail a replacement password that will confirm your membership.

Downloading a file

Once you've signed on, type *go cobbapp* at the prompt to access The Cobb Group Online. When the Cobb Applications Forum menu appears, enter *3* to select LIBRARIES (Files). Then, choose the appropriate library from the Libraries menu. For instance, to select the library containing *Inside DOS* programs and batch files, enter *4* for DOS Tools/Resources.

When the next set of options appears, you select *2* (DIRECTORY of Files) to view a list of available files or *4* (DOWNLOAD a file to your Computer) to start downloading. If you choose to view the directory, press [Enter] after each screenful to display the next page of file listings. After the last directory page, the last set of options reappears.

When you choose option 4, the next prompt asks you for a filename. Each filename represents the issue in which the batch files and programs you want to download appeared. (All batch files and programs in an issue are compressed into a single downloadable file.) Appropriate filenames start with a three-letter code for the journal followed by the last two digits of the year, a hexadecimal number representing the month (1-9 represent January-September; A-C represent October-December), and a ZIP extension. To download the batch files and programs in this issue of *Inside DOS*, for example, you'd enter the filename *IDO93C.ZIP*. The three-letter code *IDO* represents *Inside DOS*, *93* is the year, and *C*, the hexadecimal equivalent of 12, represents the month of December.

After you enter the filename, a menu of download protocols will appear. Select one that your modem supports. Then choose the command from your communications program to initiate the transfer. Once downloading is complete, press [Enter] to return to the DOS Tools/Resources options.

Before you can use batch files and programs from The Cobb Group Online, you must extract them from the compressed file. You can do so by using the PKUNZIP expansion utility. The next section describes how you can obtain PKUNZIP from the ZiffNet Library. If you already have PKUNZIP, skip to the section that follows.

Obtaining PKUNZIP

Since PKUNZIP is part of a compressed file in the main ZiffNet Library, not in The Cobb Group Online, you must first exit The Cobb Group Online by typing *top* at the current prompt and pressing [Enter]. When the main ZiffNet menu appears, select *2* to enter the Software and Utility Library. At the next set of options, choose *4* (Download a Utility). Then, type *pkz110.exe* at the *Enter file name* prompt and press [Enter]. Continue downloading as we described in the previous section. When the transfer is complete, press [Enter] to return to the ZiffNet Library menu. Finally, type *exit* and press [Enter] to log off ZiffNet.

PKZ110.EXE is a self-extracting file containing PKUNZIP as well as the PKZIP compression utility and some other files. To extract these files, simply type *pkz110* and press [Enter]. PKUNZIP will then be ready to expand compressed files downloaded from The Cobb Group Online. (If you want access to PKUNZIP from any directory, make sure you store it in a directory on your path.)

Using PKUNZIP to expand compressed files

To use PKUNZIP, simply issue the command

```
C:\>pkunzip path\filename destination
```

where *path* is the location of the compressed file, *filename* is the name of the compressed file, and *destination* is the directory you want to store the extracted files in. For example, suppose you downloaded the IDO93C.ZIP file to the C:\TEMP directory. To expand this file and store the batch files and programs it contains in the C:\BATCH directory, you'd issue the command

```
C:\>pkunzip c:\temp\ido93c.zip c:\batch
```

After PKUNZIP displays copyright information, you'll see

```
Searching ZIP: c:/IDO93C.ZIP
 Extracting: c:/batch/LETITSNO.BAS
UnShrinking: c:/batch/TODAY.BAT
  Exploding: c:/batch/ALTCOPY.BAT
```

or something similar. You can now execute these files just as if you'd created them from scratch.

DOS Software Connection

Are you on the lookout for good DOS shareware, freeware, and public domain software? If so, then you may want to subscribe to DOS Software Connection. DOS Software Connection is a service that provides you with a monthly disk loaded with useful DOS utilities, applications, games, and much, much more.

You can purchase a one-year subscription to DOS Software Connection for $59. Or, if you don't want to subscribe but would like to purchase a single disk, you can do so for only $7.50. To subscribe to DOS Software Connection or to order a single disk, just call Customer Relations at (800) 223-8720. Outside the US, please call (502) 491-1900.

VERSIONS
5.0 & 6.0

Buttons for DOS! offers user-friendly access to your non-Windows programs

Although Microsoft Windows is quickly growing into the PC environment of choice, many programs can't execute from the Windows Program Manager—programs such as Disney's Stunt Island and Psygnosis' Lemmings, as well as the MS-DOS DEFRAG and DoubleSpace utilities. In fact, many games, DOS utilities, and educational software packages can't run in a multitasking environment such as Windows.

If you've grown accustomed to the user-friendly interface that Windows provides, making the sudden transition from the Windows environment to DOS when you want to run such programs can be frustrating and tedious. To smooth this transition, Triad Software created *Buttons for DOS!*—a graphical-style DOS menu system designed to respond with the ease of Windows. When you have to exit Windows to run a non-Windows program, *Buttons for DOS!* takes over to provide the same "point-and-click" simplicity.

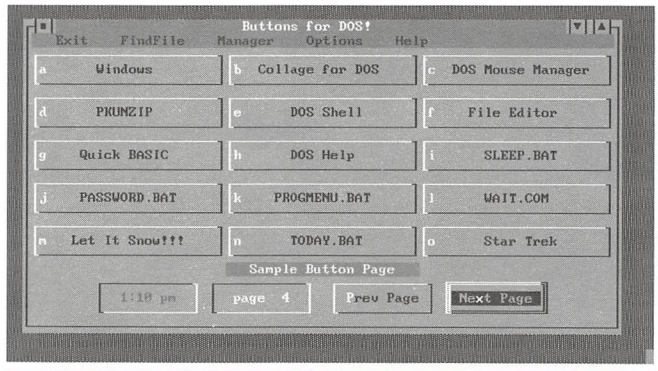
A *Buttons for DOS!* overview

Buttons for DOS! is probably one of the easiest-to-use DOS menu systems available. It allows you to assign virtually any DOS program to a button, relieving you of the task of remembering what command you have to type at the DOS prompt to execute the program. With *Buttons for DOS!*, you can create up to ten pages of buttons to run up to 150 programs that you execute just by clicking the appropriate button. Figure A shows a sample button page. When you exit a program run from the *Buttons for DOS!* menu, the menu reappears to let you run another program with the same ease. You can even restart Windows from the *Buttons for DOS!* menu.

Pulldown menus to make setup easy

To help you set up button pages and customize the program, *Buttons for DOS!* supplies easy-to-follow pulldown menus. On the Manager menu, you'll find everything you need to assign programs to buttons and arrange the buttons. For programs you want to run consecutively by clicking a single button or for those that use option switches, this menu lets you set up the button by creating a macro. From the Options menu, you can choose a command to customize the way your system loads *Buttons for DOS!*, as well as commands to tailor the screen. One option even lets you select a character to enhance the look of the mouse pointer. The FindFile feature helps you find the programs you want to assign to buttons. And the Help menu offers assistance on anything you want to know about the *Buttons for DOS!* program.

Figure A



Buttons for DOS! *lets you organize your programs on ten button pages.*

The ease of use and setup alone makes *Buttons for DOS!* a worthy investment. But Triad Software didn't stop there. In addition to being useful and user-friendly, *Buttons for DOS!* offers extras that enhance the program's value.

Quick and easy installation

Unlike many programs that take several minutes to install, *Buttons for DOS!* installs in less than 60 seconds. Its unique installation program checks what version of DOS you're running and gives you the option of automatically setting up a page of buttons for the built-in DOS utilities. The only other information you must provide is how you want your system to load *Buttons for DOS!* during startup.

Keyboard access for non-mouse users

If you don't use a mouse, you can still take advantage of the user-friendliness of *Buttons for DOS!*. The program provides easy keyboard access to all its buttons, menus, and commands. [Alt]-key combinations and arrow keys let you make selections and navigate the screen as easily as if you were using a mouse.

Built-in screen saver

A third extra that enhances *Buttons for DOS!* is the built-in screen saver—a feature that isn't commonly found in DOS-based applications. Because you're likely to want the *Buttons for DOS!* menu available at your fingertips, the screen saver makes this possible without allowing the stationary screen elements to cause screen burn-in.

System requirements

At a minimum, you need 270 Kb of free hard disk space to install and run *Buttons for DOS!*. If you're short on disk space, you can delete the FindFile feature (about 190 Kb)

without affecting the other features of *Buttons for DOS!*. In addition, your system must use MS-DOS Version 3.3 or higher.

Buttons for DOS! pricing

The retail cost of *Buttons for DOS!* is only $29. However, when you order directly from Triad Software, you can get the program for $20, postage paid. To order direct, send a check or money order for $20 to the following address:

Triad Software
P.O. Box 1299-23
Sequim, WA 98382

VAN WOLVERTON

VERSIONS
5.0 & 6.0

Creating your own DOS command shortcuts

Most application programs display some form of online help when you press [F1]. Why doesn't DOS? Furthermore, many application programs—even the DOS Shell—use [Ctrl] and [Alt] key combinations to give you shortcuts to commands. Again, why doesn't DOS?

Well, if you'd like to have those features in DOS, it's time to take matters into your own hands. All it takes is a few PROMPT commands to redefine the meaning of a key. By redefining a key to produce the name of a command or program followed by a carriage return, you can set up shortcuts to execute DOS commands, execute commands you create using either a DOS batch file or a DOSKEY macro, or run any program. The process is fairly simple:

1. Save the current command prompt definition.
2. Use a PROMPT command to execute an ANSI.SYS command that changes the result of pressing a key.
3. Restore the original DOS command prompt definition.

In order to use ANSI.SYS commands, your CONFIG.SYS file has to include the following DEVICE directive:

```
device=c:\dos\ansi.sys
```

If you're going to follow the examples in this article, make sure your CONFIG.SYS file contains this DEVICE directive; if it doesn't, add it (by editing CONFIG.SYS in the DOS Editor) and restart DOS.

ANSI.SYS commands

Located in the DOS directory, ANSI.SYS is a device driver that lets you control the keyboard and display. (These two devices make up what DOS calls the console, or CON.) When you load ANSI.SYS, it allows DOS to recognize certain character strings sent to the display as commands; instead of displaying the commands, DOS carries them out.

ANSI.SYS commands let you control your display's color and other attributes and the effect of pressing almost any key on the keyboard. There's a catch, of course: Each ANSI.SYS command begins with a character called the Escape character ({ESC}) that you can't type. (If you press the [Esc] key, DOS just erases what you've typed and waits for you to continue typing.)

However, there are two ways to produce the Escape character: You can create it in a file by using a special key combination, or you can represent it in a PROMPT command by typing *$e*. By using a PROMPT command, you can execute an ANSI.SYS command from the DOS prompt or a batch file. (See "Executing ANSI.SYS Commands with the PROMPT Command" later in this article.)

All ANSI.SYS commands start with the Escape character and a left bracket and end with a letter that identifies the command. Parameters that specify the command's particular action come in between. The ANSI.SYS command to redefine the keyboard has two such parameters and ends with a lowercase p:

```
{ESC}[key code;new resultp
```

where {ESC} is the Escape character (ASCII code 27), *key code* is the numeric code that identifies the key you want to redefine, and *new result* is what you want to happen when you press the key. You must use the PROMPT command to enter the Escape character from the command line; otherwise, you must place the Escape character and redefine the key in a file.

A character's ASCII code serves as its key code for each key on the standard typewriter portion of the

keyboard (letters, numbers, and punctuation marks). An *extended key code*—0, followed by a semicolon, followed by a second number—identifies each of the remaining keys (function keys and most [Ctrl] and [Alt] key combinations). Starting with Version 6, these codes no longer appear in the manual that comes with DOS; they're only in the *Technical Reference*, which you must buy separately. Table A lists the key codes and extended key codes you're most likely to use to create command shortcuts.

For *new result*, you can supply another key code, a command you want the key to execute, or simply a series of characters to type. If you want the key to produce a command or typed characters, just enclose *new result* in quotation marks. You can even specify a combination of key codes and character strings, as long as you separate each element with semicolons. And if you include the ASCII code for a carriage return (13) in *new result*, you won't have to press [Enter] to execute the command after pressing the redefined key.

Table A: *Key codes and extended key codes*

Key Combination	Key Code
[Ctrl]A-[Ctrl]Z	1-26
[Ctrl][	27
[Ctrl]]	28
[Ctrl]\	29
[F1]-[F10]	0;59-0;68
[Shift][F1]-[Shift][F10]	0;84-0;93
[Ctrl][F1]-[Ctrl][F10]	0;94-103
[Alt][F1]-[Alt][F10]	0;104-0;113
[Alt]A	0;30
[Alt]B	0;48
[Alt]C	0;46
[Alt]D	0;32
[Alt]E	0;18
[Alt]F	0;33
[Alt]G	0;34
[Alt]H	0;35
[Alt]I	0;23
[Alt]J	0;36
[Alt]K	0;37
[Alt]L	0;38
[Alt]M	0;50
[Alt]N	0;49
[Alt]O	0;24
[Alt]P	0;25
[Alt]Q	0;16
[Alt]R	0;19
[Alt]S	0;31
[Alt]T	0;20
[Alt]U	0;22
[Alt]V	0;47
[Alt]W	0;17
[Alt]X	0;45
[Alt]Y	0;21
[Alt]Z	0;44

Note that you separate *new result* from *key code* with a semicolon but that you don't need a semicolon between *new result* and the concluding *p*. Finally, to restore the default meaning of a key, you use the same ANSI.SYS command syntax, specifying the original key code for both *key code* and *new result*.

Some sample ANSI.SYS commands

All this may sound much more complicated than it really is. Let's look at some examples. As Table A shows, the key code for [Ctrl]V is 22, so the following ANSI.SYS command redefines the [Ctrl]V key combination to execute the VER command:

```
{ESC}[22;"ver";13p
```

Once you execute this ANSI.SYS command, you simply press [Ctrl]V anytime you want to issue the VER command.

Now let's look at an ANSI.SYS command for a key having an extended key code. Table A shows that the extended key code for [Shift][F10] is 0;93. The following ANSI.SYS command redefines [Shift][F10] to execute a MEM command with the /C and /P switches:

```
{ESC}[0;93;"mem /c /p";13p
```

Notice that in both of the examples above, we use a combination of a command (characters in quotation marks) and a key code (13 to execute a carriage return after the command) in *new result*. A semicolon separates the two, but no semicolon appears before the final *p*.

The following ANSI.SYS commands restore [Ctrl]V and [Shift][F10] to their original meanings:

```
{ESC}[22;22p
{ESC}[0;93;0;93p
```

Notice that in the second command, semicolons separate all the numbers, but again, no semicolon precedes the final *p*.

Executing ANSI.SYS commands with the PROMPT command

The simplest way to enter ANSI.SYS commands is with a PROMPT command, because it lets you use $e to represent the Escape character. The following PROMPT command carries out the ANSI.SYS command shown earlier that redefines [Shift][F10] to carry out a MEM command:

```
prompt $e[0;93;"mem /c /p";13p
```

To restore the default command prompt after issuing this command, you type

```
prompt
```

Now press [Shift][F10]. When you do, DOS displays the first screen of output from the MEM command. Press any key to display the second screen and return to the command prompt.

To restore the original meaning of [Shift][F10], type the PROMPT commands

```
prompt $e[0;93;0;93p
prompt
```

Redefining [F1] to display DOS online Help
Now you're ready to make [F1] display the DOS online Help index. Type the commands

```
prompt $e[0;59;"help";13p
prompt
```

and press [F1]. The Command Reference screen of DOS' online Help appears. From this screen, you can choose any Help topic. To return to DOS, choose Exit from the File menu.

Next, let's look at how you save and restore a custom prompt. If you've defined your own command prompt, you'll have to either type the PROMPT command required to set it up or save the prompt definition and restore it after redefining the key. The simplest way to do this is by using environment variables.

Saving and restoring the command prompt

To save the command prompt, use the SET command to define an environment variable (call it PSAV) whose value is the current prompt definition:

```
set psav=%prompt%
```

The percent signs tell DOS to replace *%prompt%* with the value of the PROMPT environment variable, which is the current command prompt definition. Now when you change the prompt definition by typing a PROMPT command to redefine a key, the PSAV environment variable contains the original prompt definition. You can restore your command prompt with another SET command:

```
set prompt=%psav%
```

This reverses the process of the first SET command, restoring your command prompt. The following three-command sequence redefines [F1] to display Help, leaving your command prompt undisturbed:

```
set psav=%prompt%
prompt $e[0;59;"help";13p
set prompt=%psav%
```

To restore the original meaning of [F1], you just specify the key code as both the old and new result:

```
set psav=%prompt%
prompt $e[0;59;0;59p
set prompt=%psav%
```

Solving a potential problem

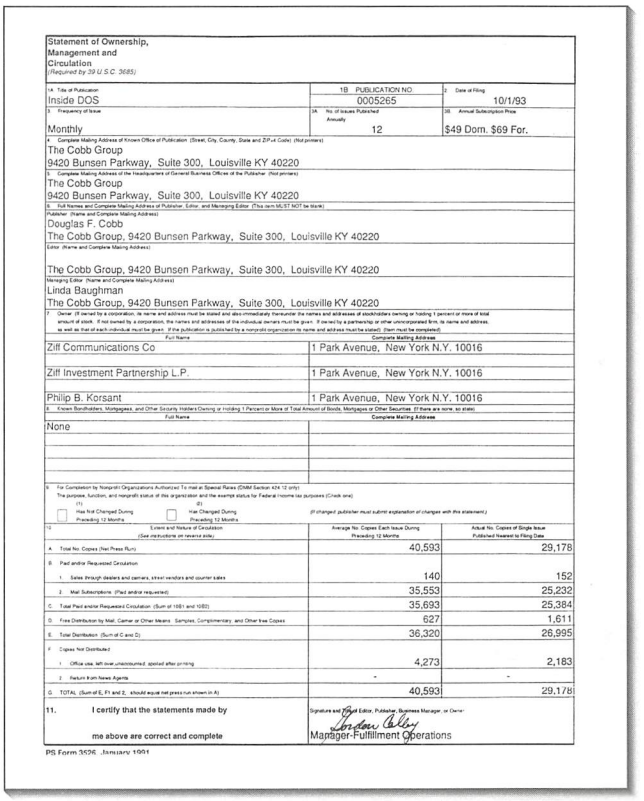
You can get into a spot of trouble messing around with the keyboard this way. For example, suppose you accidentally redefined the 0 key to issue the DIR command. That probably isn't what you wanted to do, because there are times when you'll want to type *0*.

You can't use the PROMPT command to restore the default meaning of the 0 key because the command—*prompt $e[0;48;0;48p*—contains 0's and every time you press 0, you'll get *dir* instead. It seems like you can't get there from here.

But most application programs ignore keyboard redefinitions, so you can solve your dilemma without restarting DOS: Just use the DOS Editor to write a one-line batch file that contains the aforementioned PROMPT command and execute the batch file.

Creating command shortcut keys is a task tailor-made for batch files. Next month, we'll look at a couple of batch files that take most of the tedium out of typing those PROMPT commands. Then we'll look at how to call these batch files from another batch file to automate the entire process.

Contributing editor Van Wolverton is the author of the best-selling books Running MS-DOS *and* Supercharging MS-DOS. *Van, who has worked for IBM and Intel, lives in Alberton, Montana.*



Statement of Ownership, Management and Circulation
(Required by 39 U.S.C. 3685)

1A. Title of Publication	1B. PUBLICATION NO.	2. Date of Filing
Inside DOS	0005265	10/1/93
3. Frequency of Issue	3A. No. of Issues Published Annually	3B. Annual Subscription Price
Monthly	12	$49 Dom. $69 For.

4. Complete Mailing Address of Known Office of Publication (Street, City, County, State and ZIP+4 Code) (Not printers)
The Cobb Group
9420 Bunsen Parkway, Suite 300, Louisville KY 40220

5. Complete Mailing Address of the Headquarters of General Business Offices of the Publisher (Not printers)
The Cobb Group
9420 Bunsen Parkway, Suite 300, Louisville KY 40220

6. Full Names and Complete Mailing Address of Publisher, Editor, and Managing Editor (This item MUST NOT be blank)
Publisher (Name and Complete Mailing Address)
Douglas F. Cobb
The Cobb Group, 9420 Bunsen Parkway, Suite 300, Louisville KY 40220
Editor (Name and Complete Mailing Address)
The Cobb Group, 9420 Bunsen Parkway, Suite 300, Louisville KY 40220
Managing Editor (Name and Complete Mailing Address)
Linda Baughman
The Cobb Group, 9420 Bunsen Parkway, Suite 300, Louisville KY 40220

7. Owner (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total amount of stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given. If the publication is published by a nonprofit organization its name and address must be stated) (Item must be completed)

Full Name	Complete Mailing Address
Ziff Communications Co	1 Park Avenue, New York N.Y. 10016
Ziff Investment Partnership L.P.	1 Park Avenue, New York N.Y. 10016
Philip B. Korsant	1 Park Avenue, New York N.Y. 10016

8. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages or Other Securities. (If there are none, so state)

Full Name	Complete Mailing Address
None	

9. For Completion by Nonprofit Organizations Authorized To mail at Special Rates (DMM Section 424.12 only)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes (Check one)
(1) ☐ Has Not Changed During Preceding 12 Months ☐ Has Changed During Preceding 12 Months (2) (If changed, publisher must submit explanation of changes with this statement.)

10. Extent and Nature of Circulation (See instructions on reverse side)	Average No. Copies Each Issue During Preceding 12 Months	Actual No. Copies of Single Issue Published Nearest to Filing Date
A. Total No. Copies (Net Press Run)	40,593	29,178
B. Paid and/or Requested Circulation		
1. Sales through dealers and carriers, street vendors and counter sales	140	152
2. Mail Subscriptions (Paid and/or requested)	35,553	25,232
C. Total Paid and/or Requested Circulation (Sum of 10B1 and 10B2)	35,693	25,384
D. Free Distribution by Mail, Carrier or Other Means. Samples, Complimentary, and Other Free Copies	627	1,611
E. Total Distribution (Sum of C and D)	36,320	26,995
F. Copies Not Distributed		
1. Office use, left over, unaccounted, spoiled after printing	4,273	2,183
2. Return from News Agents	-	-
G. TOTAL (Sum of E, F1 and 2. should equal net press run shown in A)	40,593	29,178

11. I certify that the statements made by me above are correct and complete

Signature and Title of Editor, Publisher, Business Manager, or Owner
Manager-Fulfillment Operations

PS Form 3526, January 1991

VERSIONS 5.0 & 6.0

Another way to guard against the hidden DOS trap

Richard M. Arias, a senior at Central Catholic High School in San Antonio, Texas, developed the batch file we present in this article.

In the February 1993 issue of *Inside DOS*, Van Wolverton brought to light a hidden danger in using the COPY command—the possibility of losing an existing file by copying a file and giving it the same name as the existing file. (See "Guarding Against the Hidden DOS Trap.") To provide you with protection from this pitfall, Van created SAFECOPY.BAT, a batch file that tells you when you've tried to copy a file over an existing file and then aborts the process. Many of you probably now use SAFECOPY instead of COPY every time you copy files.

But do you ever find that you intentionally want to overwrite a file by copying another one over it? If so, you could resort to using DOS' built-in COPY command. But what if you've redefined the COPY command with a Doskey COPY macro that runs the SAFECOPY batch file? Again, there's a way around this. However, the best solution is to combine the protection of SAFECOPY.BAT with an option to complete the copy process. In this article, we present the ALTCOPY batch file—an alternative to COPY and SAFECOPY that gives you this option.

Creating the ALTCOPY batch file

If you're familiar with SAFECOPY.BAT, you'll see that ALTCOPY.BAT, shown in Figure A, is very similar. To create the ALTCOPY batch file, use the DOS Editor to enter the code. You create the bullet character (•) in the third, sixth, and seventh ECHO statements by pressing [Ctrl]P, then [Ctrl]G. Once you've entered the code, save it under the name ALTCOPY.BAT.

After creating the ALTCOPY batch file, you run it by typing

```
C:\>altcopy source destination
```

and pressing [Enter]. When specifying *source* and *destination*, keep the following rules in mind:

- If you enter a complete file specification (path and filename) for *source*, enter a complete file specification for the destination as well.
- If you want to use wildcards in *source* to copy more than one file, make sure the files you're copying are in the current directory, enter *source* without a path, and enter *destination* without a filename.

Figure A

```
@echo off
rem Richard M. Arias' ALTCOPY.BAT warns you when you're
rem about to overwrite an existing file by copying another
rem on top of it. It then gives you the option of continuing
rem the copy process or aborting it.
echo.
if not "%2"=="" goto :OK1
echo      •You must enter a destination
echo.
echo ---FILE NOT COPIED---
goto :END

:OK1
if exist %1 goto :OK2
echo      •%1 does not exist
goto :END

:OK2
set file=%2
for %%a in (C:\ c:\) do if %file%==%%a set file=C:
rem Repeat the above line, substituting the appropriate
rem drive letter, for every drive on your system (including
rem floppy drives).
set dirfile=%file%\%1
if exist %file% goto :EXISTS
if exist %dirfile% goto :EXISTS
goto :COPI

:EXISTS
if exist %dirfile% set dest=%dirfile%
if exist %file% set dest=%file%
echo       •The destination file already exists. Do you wish to
echo        overwrite it?
echo        Press Y for yes or N for no
erase %dest% /p > nul
if exist %dest% echo --FILE NOT COPIED--
if exist %dest% goto :END

:COPI
copy %1 %2

:END
set dest=
set file=
set dirfile=
echo.
```

ALTCOPY.BAT lets you choose whether to overwrite existing files.

How ALTCOPY.BAT works

ALTCOPY.BAT, like most batch files, begins with @ECHO OFF to prevent DOS from displaying each command in the batch file. The next four REM statements explain the function of the batch file. Then, the ECHO. statement echoes a blank line to the screen.

The first IF statement

```
if not "%2"=="" goto :OK1
```

checks to see that you specified a value for *destination* (%2) when you ran the batch file. If you didn't, the batch file executes the lines

```
echo      •You must enter a destination
echo.
echo ---FILE NOT COPIED---
goto :END
```

The bullet character in the first ECHO statement tells your computer to beep as it reminds you that you must specify a destination to copy the file to. The other two ECHO statements print a blank line and tell you the copy process failed. Finally, the GOTO statement branches to the :END section of the batch file.

The :OK1 section

If you did specify a destination, control branches to the :OK1 section:

```
:OK1
if exist %1 goto :OK2
echo     •%1 does not exist
goto :END
```

The IF statement in this section checks to see that the file you want to copy (*%1*) exists. If it doesn't, the next two lines execute. The ECHO statement makes your computer beep as it displays a message that the file doesn't exist, and the GOTO statement branches to the :END section of the batch file.

The :OK2 section

If the file you want to copy does exist, control branches to the :OK2 section:

```
:OK2
set file=%2
for %%a in (C:\ c:\) do if %file%==%%a set file=C:
rem Repeat the above line, substituting the appropriate
rem drive letter, for every drive on your system (including
rem floppy drives).
set dirfile=%file%\%1
if exist %file% goto :EXISTS
if exist %dirfile% goto :EXISTS
goto :COPI
```

The first SET statement creates the *file* environment variable and assigns it the value you specified as *destination*. The FOR statement checks to see if the destination you specified is the root directory of the C: drive. (The REM statements simply remind you to create similar FOR statements to check for the root directories of all your drives.)

If the destination you specified is a root directory, the batch file changes the value of *file* to include just the drive specification. It then concatenates the value of *file* and the value of *source* to create a complete file specification in the *dirfile* environment variable:

```
set dirfile=%file%\%1
```

(If the destination you specified is a root directory, the format of this statement requires that *file* be just a drive specification.) If the destination you specified is just a directory, *dirfile* identifies the complete file specification the copied file will have.

Next, the batch file checks to see if a file having the same name as the destination file exists. The first IF EXIST statement screens instances where you provided a complete file specification for the destination. The second IF EXIST statement checks instances where you provided a directory as the destination. Both IF EXIST statements branch control to the :EXISTS section if a file with the same name exists; otherwise, control branches to the :COPI section.

The :COPI section

The single statement in the :COPI section

```
copy %1 %2
```

simply issues DOS' built-in COPY command to copy the file you specified as *source* to the *destination* directory or file. Once DOS copies the file, the statements in the :END section execute.

The :EXISTS section

If control branches to the :EXISTS section, the first two statements

```
if exist %dirfile% set dest=%dirfile%
if exist %file% set dest=%file%
```

store the name of the existing file in the *dest* environment variable. (Since the condition in only one of these IF EXIST statements can be true, only one will execute.) The next ECHO statements

```
echo     •The destination file already exists. Do you wish to
echo      overwrite it?
echo      Press Y for yes or N for no
```

tell your computer to beep, indicate that a file matching the destination you specified already exists, and ask whether you want to overwrite this file. Then the statement

```
erase %dest% /p > nul
```

issues the ERASE command with the destination file as its object and the /P switch to prompt for confirmation. Since the preceding ECHO command already prompts you to press Y or N, the ERASE prompt is unnecessary and is therefore redirected to the NUL device. Note that this command will execute and wait for a response long before you've finished reading the ECHO prompt.

The last two lines in the :EXISTS section

```
if exist %dest% echo --FILE NOT COPIED--
if exist %dest% goto :END
```

check whether you decided to overwrite the existing file.

Microsoft Technical Support
(206) 454-2030

If you pressed N to prevent overwriting, the destination file still exists. The message --*FILE NOT COPIED*-- will appear on your screen, and the batch file will branch to the :END section. If you pressed Y to overwrite the existing file, the batch file will continue with the :COPI section.

The :END section

Regardless of how the batch file proceeds, the last section to execute is always the :END section:

```
:END
set dest=
set file=
set dirfile=
echo.
```

This section simply clears the values of the *dest*, *file*, and *dirfile* environment variables, prints a final blank line, and ends the batch file.

Notes and recommendations

Before using ALTCOPY.BAT, you should note the following:

- Since ALTCOPY.BAT erases an existing file before copying another file to the same name, you can recover the erased file by using the UNDELETE utility. (For more on UNDELETE, see the articles "DOS 6's UNDELETE Offers Three Levels of Protection from File Deletion" and "Undeleting Files in DOS 5," in the August 1993 issue.) For best results, activate the MIRROR feature in DOS 5 or Delete Tracker or Delete Sentry in DOS 6 for maximum protection from premature file deletion.
- As the code in Figure A shows, ALTCOPY.BAT creates several temporary environment variables. Therefore, you might need to increase the environment size defined in your CONFIG.SYS file. To do so, make sure the line

  ```
  shell=c:\dos\command.com c:\dos\ /e:300 /p
  ```

  appears in your CONFIG.SYS file. (Your COMMAND.COM path may be different.) The /E:300 switch increases the size of the environment to 300 bytes. You can adjust this number to suit your needs. ■

LETTERS

VERSION 6.0

Pressing an invalid key suspends CHOICE's timed default

In the October 1993 issue of *Inside DOS*, you suggested adding the lines

```
echo message
choice /t:y,15 > nul
```

to a batch file when you want to display a message onscreen for 15 seconds. (See "Displaying a Batch File Screen for a Specific Time.") Although this technique is quite useful, it isn't foolproof. If you press a key other than Y or N (CHOICE's default valid keys) during the pause, CHOICE ignores the /T switch. Therefore, the command won't default to the key you specified (in this case, Y) and will wait indefinitely for you to press a valid key. Likewise, suppose you use the /C switch to specify a valid key or set of keys, such as

```
choice /c:z /t:z,15 > nul
```

instead of using CHOICE's defaults. If you then press an invalid key during the pause (in this case, any key except Z), CHOICE will pause until you press the valid key, no matter how long it takes you.

You could work around this limitation by validating most of the keyboard in the /C switch. However, a few regular keys (such as [Spacebar], [Tab], /, ¦, <, and >) and some function keys will still cause CHOICE to hang. Your best bet is to keep this limitation in mind and remember not to press an invalid key during CHOICE's pause.

Gil Bowers
Menlo Park, California

Mr. Bowers brings to light an important warning we missed when we presented the technique. We thank him for catching our oversight. ■

Printed in the USA
Printed on recycled and recyclable paper
50% recylced fiber including 10% post-consumer waste paper